2B2-1

類似部分森が頻出するパターン森の発見

Finding Subforest Patterns with Frequent Occurrence of Similar Subforests

戸坂 央　　中村 篤祥　　工藤 峰一
Hisashi Tosaka　Atsuyoshi Nakamura　Mineichi Kudo

北海道大学大学院 情報科学研究科
Graduate School of Information Science and Technology, Hokkaido University

We study a novel problem of mining subforest patterns with frequent occurrence of similar subforests, and propose an algorithm for this problem. In our problem setting, frequency of a pattern is counted not only for equivalent subforests but also for similar subforests. In order not to doubly count essentially the same parts of a tree, we adopt the notion of locally optimal similar pairs of subforests. Our algorithm runs in time $O(|T|^3)$, where $|T|$ is the number of nodes in a given tree T.

1. はじめに

ラベル付き順序木構造データは Web マイニング，ネットワーク解析，バイオインフォマティクス等，多くの分野で扱われている．これらのラベル付き木データからのマイニング手法として頻出部分木マイニングが近年盛んに研究されている [Asai 02, Chi 05, Zaki 05]．頻出部分木マイニングにおいて有用なパターンを発見するためには，構造の揺らぎやノイズを考慮することが必要である．これまでは主に，柔軟なマッチングを許す事により，完全に一致する構造で有用なパターンを発見する研究が行われてきた [Asai 02, Zaki 05]．それに対し，著者らはマッチングを柔軟にするのではなく，完全一致基準を緩める事により，類似する部分木も数え上げる頻出パターン木発見問題を扱い，この問題を解くアルゴリズムを提案していた [Tosaka 07]．Web ページからのデータレコード抽出 [Liu 03, Zhai 05, Zhao 05] の予備実験では，提案手法が領域推定問題において，ある程度有効である事が確かめられた．

本稿では，パターンのクラスを木から森へ拡張した問題，つまり，類似部分森が多く存在するパターン森を発見する問題を扱う．類似構造の数え上げは，共通部分を多く含む本質的には同じ部分を何度もカウントしてしまうという問題がある．我々はこの問題に対処し，より直感にあったパターン森とその出現の対を得る為に，2 つの部分森の局所最適類似の概念を導入する．そして，この問題を効率的に解くアルゴリズムを提案する．提案アルゴリズムは，文書からパターン文字列の最類似部分文字列を抽出するアルゴリズムを応用したもので，与えられた木 T に類似部分森が多く存在するパターン森を時間計算量 $O(|T|^3)$ で列挙する．但し，$|T|$ は木 T のノード数とする．

2. 問題設定

2.1 ラベル付き順序森

本稿では，主にラベル付き順序木とラベル付き順序木で構成された森である**ラベル付き順序森**を扱い，これらを単に木と森と表記する．

木 T において，ノード v と u が同じ親ノードをもち，v の方が u より先にあるとき $v \leq u$ と表記する．また，v の先頭の子ノードを f_v，最後尾の子ノードを l_v で表す．木 T において，ノード v を根とする，v の全ての子孫を持つ木を「T の v を根とする**部分木**」と呼び T_v と表記する．また，木 T において，v の子ノード i, j の間にある子ノード $k (i \leq k \leq j)$ を根とする部分木列を，「T のノード i から j を根として持つ**部分森**」と呼び，$F_v(i, j)$ と表記する．特に，$F_v(f_v, l_v)$ を「v を親に持つ部分森」と呼び単に F_v と表す．他の表記法として，森 F_1 の後ろに森 F_2 を配置することによってできる森を $F_1 \bullet F_2$ と表記する．構成する部分木の数が 0 の森を**空森** $\emptyset$ と表す．

2.2 森構造間の距離関数

提案手法では森構造間の距離として，構成木の列を文字列と見なした文字編集距離 D_S を使う．これは，木構造間の距離 d が与えられたとき，以下のような漸化式で定義される．

$D_S(\emptyset, \emptyset) = 0,$

$D_S(F_1 \bullet T_1, F_2 \bullet T_2) =$

$$\min \begin{cases} D_S(F_1, F_2) + d(T_1, T_2), \\ D_S(F_1 \bullet T_1, F_2) + d(\emptyset, T_2), \\ D_S(F_1, F_2 \bullet T_2) + d(T_1, \emptyset). \end{cases}$$

注意) 木構造間の距離の定義は，森構造間の距離の定義を含んでいると考えられるものが多い．ここでは使用する木構造間の距離 d の定義から自然に導かれる森構造間の距離を使用しなかった．D_S を用いる事は，構成要素まで似ている事を重視し，複数の構成要素が 1 つの構成要素に対応するような類似は考慮しない事に対応する．

2.3 森の近似出現

$k (\geq 0)$ を距離閾値，$D(F_1, F_2)$ を森 F_1, F_2 間の距離とする．ここで，$D(F_1, F_2) \leq k$ を満たす F_2 を F_1 の**近似出現**と呼ぶ．本稿では，森の距離関数に D_S を用いる．

2.4 局所最適なパターン森と近似出現の対

近似出現全てを数え上げると，実際には 1 つの出現とみなすべきものを複数回数え上げてしまう可能性がある．

例 1 図 1 では，距離閾値を $k = 2$ としたとき，$D_S(F_1, F_2) = 1, D_S(F_1, F_2') = 2$ で，F_2 の他にも F_2' が F_1 の近似出現となる．F_2' は F_1 の近似出現 F_2 に余分なノードが付いたものと考えられるので，F_2 と F_2' を別のものとして数え上げるのは好ましくない．

連絡先: 戸坂 央, 北海道札幌市北区北 14 条西 9 丁目北海道大学大学院 情報科学研究科コンピュータサイエンス専攻 数理計算科学講座情報認識学研究室, 011-706-6854, t_hisashi@main.ist.hokudai.ac.jp

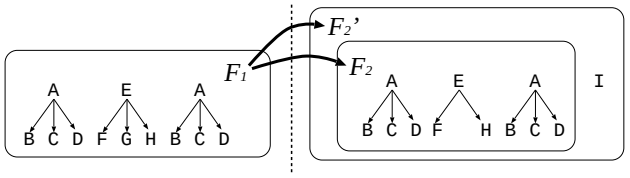

図 1: 近似出現の重複カウント

そこで，パターン森の出現カウントをより直感に即したものにするため，局所最適類似部分森対の概念を取り入れる．

以下の条件を満たす木 T の部分森 $F_{v_1}(l,t)$ を森 F の**局所最適**な近似出現と呼ぶ．また，(1),(3) を満たす $F_{v_1}(l,t)$ を F の左側局所最適近似出現と呼ぶ．

- 互いに近似出現である

$$D(F, F_{v_1}(l,t)) \leq k \tag{1}$$

- 同一の階層における局所最適性
  $\forall t' \geq l$,

$$D(F, F_{v_1}(l,t)) \leq D(F, F_{v_1}(l,t')) \tag{2}$$

  $\forall l' \leq t$,

$$D(F, F_{v_1}(l,t)) \leq D(F, F_{v_1}(l',t)) \tag{3}$$

- 異なる階層における局所最適性

$$D(F, F_{v_1}(l,t)) \leq D(F, T_{v_1}) \tag{4}$$

$l \leq \forall v \leq t, f_v \leq \forall i \leq \forall j \leq l_v$

$$D(F, F_{v_1}(l,t)) \quad \leq \quad D(F, F_v(i,j)) \tag{5}$$

ここで，互いに局所最適な近似出現である部分森の対を**局所最適類似部分森対**[*1] と呼ぶ．

例 2 図 2 の 2 つの森 P, F を用いて局所最適類似部分森対の例を示す．距離の閾値 $k = 2$ のとき，P の部分森 F_1 の F における近似出現は F_2, F_2', F_3, F_3' の 4 つ存在する．また，F_1' の近似出現は F_2, F_2', F_3 の 3 つである．これらの部分森対のうち，(F_1, F_2) は $D_S(F_1, F_2) \leq D_S(F_1, F_2')$ かつ $D_S(F_1, F_2) \leq D_S(F_1', F_2)$ で，式 (1)〜(5) を全て満たす．ゆえに，(F_1, F_2) は局所最適類似部分森対である．また，$(F_1, F_3), (F_1', F_2')$ も同様に，式 (1)〜(5) を全て満たすため，局所最適類似部分森対である．

2.5 近似頻出パターン森

$\sigma(\geq 1)$ を最小サポートとする．森 F を要素として含む木 T の局所最適類似部分森対の数が σ 以上のとき，F を T における**近似頻出パターン森**と定義する．

[*1] 条件を (1),(2),(3) に限れば，文字列に対する局所最適性が定義される．この定義と，[Erickson 83] で定義された局所最適性の違いは，[Erickson 83] では類似性が高いほど値が大きくなるスコアで定義されているため，不等号が逆であるということの他，条件 (2),(3) では 2 つの文字列同時に端を動かした場合までは最適性を要求していないということが挙げられる．

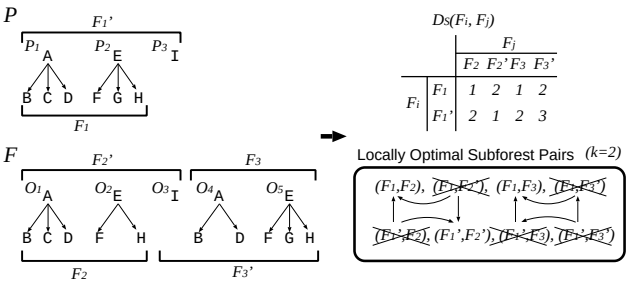

図 2: 局所最適類似部分森対の例

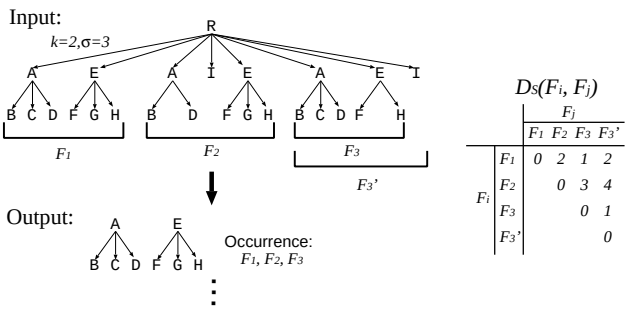

図 3: 近似頻出部分森発見問題の例

2.6 近似頻出パターン森発見問題

類似部分森が頻出するパターン森の発見問題を以下のように定義する．

入力: 木 T，距離閾値 $k(\geq 0)$，最小サポート $\sigma(\geq 1)$

出力: T の部分森であり，かつ T における近似頻出パターン森であるもの全て

例 3 図 3 に近似頻出パターン森発見問題の入出力例を示す．図のような木と距離閾値 $k = 2$，最小サポート $\sigma = 3$ が与えられたとする．与えられた木の部分森 F_1, F_2, F_3, F_3' について考える．それぞれを要素として持つ局所最適類似部分森対を列挙すると，$(F_1, F_1), (F_1, F_2), (F_1, F_3), (F_2, F_2), (F_3, F_3)$ の 5 つになる．$(F_1, F_3'), (F_3, F_3')$ は $D_S(F_1, F_3) < D_S(F_1, F_3'), D_S(F_3, F_3) < D_S(F_3, F_3')$ であるため，局所最適類似部分森対ではない．ゆえに，最小サポート以上の局所最適類似部分森対に含まれる F_1 を，求める近似頻出パターン森の一つとして出力する．

3. アルゴリズム

この節では類似部分森が頻出するパターン森の発見問題を効率良く解くアルゴリズムを示す．提案アルゴリズムでは，木 T が与えられたとき，以下の 3 つのステップに分けて近似頻出パターン森を求める．

Step.1 T における全ての部分木間の距離 d を求める．

Step.2 T における全ての部分森 P の (左側局所最適な) 近似出現を列挙し，P の近似出現集合 $Occ(P)$ と局所最適性のチェックテーブルを作成する．

Step.3 各部分森 P の近似出現のうち，局所最適性を満たすものを数え，近似頻出パターン森を列挙する．

以下ではそれぞれのステップの詳細について説明する．

3.1 部分木間の距離計算

Step.1 では様々な距離に対するアルゴリズムを使う事が可能である．高速なアルゴリズムとしては，Tree Constrained Editing Distance を採用すれば [Zhang 95] のアルゴリズムを用いて，$O(|T|^2)$ で求める事が可能である．

3.2 パターン森の近似出現の列挙

T における全ての部分森 (パターン森) の T における左側局所最適な近似出現を列挙する．木 T に存在する部分森は $O(|T|^2)$ 個存在するため，単純にこれら全ての部分森対において D_S を計算する際の時間計算量は $O(|T|^6)$ となり実用に耐え得ない．そこで，文書に含まれるパターン文字列の最類似部分文字列の抽出を行う動的計画法に基づくアルゴリズムを応用し，パターン森の左側局所最適な近似出現を時間計算量 $O(|T|^3)$ で列挙する．

パターン文字列 $P_S = p_1p_2 \ldots p_m$ と文書 $D = d_1 \ldots d_n$ を考える．与えられた文書からパターン文字列の最類似部分文字列を抽出するアルゴリズム [Sellers 80] では，$1 \leq i \leq m$ を満たす全ての $p_1p_2 \ldots p_i$ と，$1 \leq t \leq n$ を満たす全ての t において，$p_1p_2 \ldots p_i$ と $d_s \ldots d_t (1 \leq s \leq t)$ の距離を最小とする s，つまり，$p_1p_2 \ldots p_i$ の左側局所最適な部分文字列を計算量 $O(|P_S| \cdot |D|)$ で列挙できる．ゆえに，$1 \leq j \leq m$ を満たす全ての $p_jp_2 \ldots p_m$ と文書 D に対しアルゴリズムを適用すれば，パターン文字列の部分文字列の左側局所最適性を満たす (文書中の) 部分文字列を得る事ができる．よって，パターン文字列のすべての部分文字列に対し，左側局所最適性を満たす (文書中の) 部分文字列の列挙は時間計算量 $O(|P_S|^2 \times |D|)$ で可能である．

本稿で用いる森の距離 D_S は，森の要素である木を 1 文字，木の距離を文字置換のコストとして扱ったときの文字列の編集距離である．ゆえに，全ての部分木間の距離が既知であれば，パターン森と左側局所最適性を満たす部分森は，文字列の最類似部分文字列抽出アルゴリズム [Sellers 80] で計算できる．木 T の全ての部分森が近似頻出パターン森候補なので，T の全てのノード v_1 に対する部分森 $F_{v_1} = P_1 \bullet \ldots \bullet P_m$ における，$1 \leq j \leq m$ を満たす全ての $P_j \bullet \ldots \bullet P_m$ と，T の全てのノード v_2 に対する部分森 $F_{v_2} = O_1 \bullet \ldots \bullet O_n$ の組み合わせにおいて，最類似部分文字列抽出アルゴリズムを適用すれば，全てのパターン森の左側局所最適な近似出現が得られる．ゆえに，ここで得られるパターン森と近似出現の対は局所最適類似部分森対を全て含んでいる．

文書からパターン文字列の最類似部分文字列を抽出するアルゴリズムを応用した，パターン森の左側局所最適な近似出現を列挙するアルゴリズムを以下に示す．

Loop.1 行きがけ順に部分森 $F_{v_1} = P_1 \bullet \ldots \bullet P_m (v_1 \in T)$ を選択し，Loop.2 を繰り返す．

Loop.2 $1 \leq j \leq m$ を満たす全ての $P = P_j \bullet \ldots \bullet P_m$ について Loop.3 の作業を行う．

Loop.3 行きがけ順に部分森 $F_{v_2} = O_1 \bullet \ldots \bullet O_n (v_2 \in T)$ を選択し，P と F_{v_2} の組み合わせにおいて以下の左側局所最適な近似出現を列挙するアルゴリズムを適用する．

動的計画法で用いる $(m+1) \times (n+1)$ の大きさのテーブルにおける i 行 j 列目の要素を $D^*[i,j]$ と表す．まず，全ての

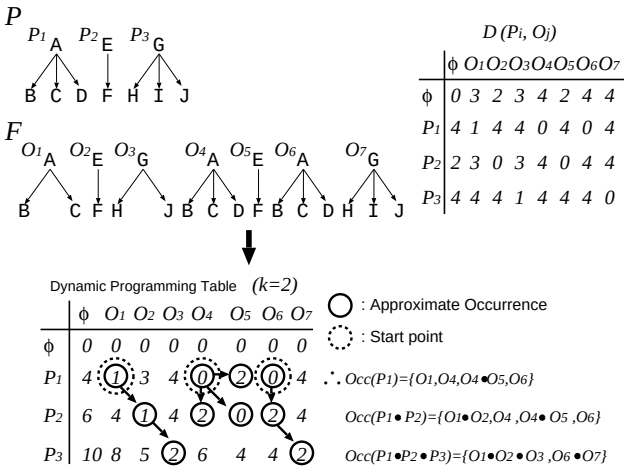

図 4: 近似出現の列挙

$0 \leq s \leq n$ を満たす s において $D^*[0,s] = 0$，全ての $1 \leq t \leq m$ を満たす t において $D^*[t,0] = D_S(P_1 \bullet \ldots \bullet P_t, \emptyset)$ と初期化する．また，近似出現の開始地点 $Start[i,j]$ を，全ての $0 \leq s \leq n, 0 \leq t \leq m-1$ を満たす s,t において $Start[s,0] = \{1\}, Start[0,t] = \{t+1\}$ と初期化する．そして，残りの m 行 n 列には以下の式による代入を行う $(1 \leq i \leq m, 1 \leq j \leq n)$.

$$D^*[i,j] \leftarrow \min \begin{cases} D^*[i-1,j-1] + D_S(P_i, O_j), & (6) \\ D^*[i-1,j] + D_S(P_i, \emptyset), & (7) \\ D^*[i,j-1] + D_S(\emptyset, O_j). & (8) \end{cases}$$

また，上の式でどの式が最小となったかによって $Start[i,j]$ に

$$\begin{cases} Start[i-1,j-1] \ (D^*[i,j] \leftarrow \text{式 (6) のとき}), \\ Start[i-1,j] \ (D^*[i,j] \leftarrow \text{式 (7) のとき}), \\ Start[i,j-1] \ (D^*[i,j] \leftarrow \text{式 (8) のとき}). \end{cases}$$

を追加する $(1 \leq i \leq m, 1 \leq j \leq n)$．ここで，$D^*[i,j] \leq k$ かつ $a \in Start[i,j] (1 \leq a \leq j)$ が成り立つとき，$O_a \bullet \ldots \bullet O_j$ は $P_1 \bullet \ldots \bullet P_i$ の左側局所最適な近似出現である．このとき，$Occ(P_1 \bullet \ldots \bullet P_i)$ に $O_a \bullet \ldots \bullet O_j$ を追加し，更に，$P_1 \bullet \ldots \bullet P_i$ と $O_a \bullet \ldots \bullet O_j$ に対し局所最適性のチェックテーブルの更新を行う (3.3 節参照).

例 4 図 4 に距離閾値を $k = 2$ としたときの，森 $F = O_1 \bullet \ldots \bullet O_7$ における，パターン森 $P = P_1 \bullet \ldots \bullet P_t (1 \leq t \leq 3)$ の (左側局所最適な) 近似出現の列挙例を示す．まず，動的計画法で用いるテーブルの P_1 の行を計算する．P_1 の近似出現は距離が閾値以下の O_1, O_4, O_6 と O_4 から拡張された $O_4 \bullet O_5$ の 4 つと分かる．次に，$P_1 \bullet P_2$ の行を計算する．$P_1 \bullet P_2$ の近似出現は O_1 から拡張された $O_1 \bullet O_2$，O_4 から拡張された $O_4 \bullet O_5$ と O_4 自身，そして O_6 となる事が分かる．最後に $P_1 \bullet P_2 \bullet P_3$ の行を計算する．$P_1 \bullet P_2 \bullet P_3$ の近似出現は，$O_1 \bullet O_2$ から拡張された $O_1 \bullet O_2 \bullet O_3$ と，O_6 から拡張された $O_6 \bullet O_7$ であることが分かる．

このアルゴリズムで作成されるテーブルのサイズの和は $O(|T|^2) \times O(|T|)$ である為，アルゴリズムの時間計算量は $O(|T|^3)$ となる．

3.3 局所最適性のチェックテーブルの更新

3.2 節で得られたパターン森と近似出現の対に対し，局所最適性の条件式 (2),(4), (5) を満たしているかチェックするために使うテーブルの更新を行う．最適性のチェックを行う為に以下の情報を格納する 3 つのテーブル (LOTmpOE，LOTmpPE, LOTmpPB) を用意する．

LOTmpOE(P_B, P_E, O_B)**:** $P_B \bullet \ldots \bullet P_E$ との距離が現時点で最小の $O_B \bullet \ldots \bullet O_E$ に対する O_E の集合 $OptO_E$ と，その距離 $DTmpO_E$ を格納．

LOTmpPE(P_B, O_B, O_E), LOTmpPB(P_E, O_B, O_E) についても同様に定義される．

パターン森 $P = P_1 \bullet \ldots \bullet P_m$ とその近似出現 $O = O_1 \bullet \ldots \bullet O_n$ に対し，テーブル LOTmpOE は以下のように更新される．

if $D_S(P, O) < DTmpO_E$ **then**
  $DTmpO_E \leftarrow D(P, O)$
  $OptOE \leftarrow \{O_n\}$
else if $D(P, O) = DTmpO_E$ **then**
  $OptO_E \leftarrow OptO_E \cup \{O_n\}$
end if

また，テーブル LOTmpPE, LOTmpPB も同様に更新する．そして，以下のように出現リストから異なる階層における局所最適性を満たさないものを削除し，3.4 節での近似頻出パターンの列挙で扱われないようにする．

v_1 を P の根の親ノード，v_2 を O の根の親ノードとする．
if $D_S(P, O) < D_S(P, T_{v_2})$ **then**
  $Occ(P) \leftarrow Occ(P) - \{T_{v_2}\}$
end if
if $D_S(P, O) > D_S(P, T_{v_2})$ **then**
  $Occ(P) \leftarrow Occ(P) - \{O\}$
end if
if $D_S(P, O) < D_S(T_{v_1}, O)$ **then**
  $Occ(T_{v_1}) \leftarrow Occ(T_{v_1}) - \{O\}$
end if
if $D_S(P, O) > D_S(T_{v_1}, O)$ **then**
  $Occ(T_{v_1}) \leftarrow Occ(P) - \{O\}$
end if

3.4 近似頻出パターン森の列挙

3.2 節によって得られたパターン森と近似出現の対と，3.3 節によって得られた局所最適性のチェックテーブルから，近似頻出パターン森を列挙する．以下の手順で局所最適類似部分森対を数え上げ，出現カウントが最小サポート σ 以上の場合にパターン森を近似頻出パターン森として出力する．

Loop.1 全てのパターン森 $P = F_v(i, j)(v \in T, f_v \leq i \leq j \leq l_v)$ において Loop.2 を繰り返す．

Loop.2 P の近似出現集合 $Occ(P)$ の全ての要素 O において，部分森対 (P, O) を 3 つの局所最適性チェックテーブルと照合し，局所最適類似部分森対であった場合に出現カウントを増やす．

4. おわりに

本論文では，直感にあった出現の数え上げを行う類似部分森が頻出するパターン森の発見問題を定義した．また，この問題を，パターン文字列の最類似部分文字列を抽出するアルゴリズムを応用して効率的に解くアルゴリズムを提案した．

今後は，提案アルゴリズムを実装し，Web やバイオインフォマティクス等の分野における実際のデータに応用し，得られるパターンの有用性を検証する．

参考文献

[Asai 02] Asai, T., Abe, K., Kawasoe, S., Arimura, H., Sakamoto, H., and Arikawa, S.: Efficient substructure discovery from large semi-structured data, in *Proceedings of the 2nd SIAM international conference on Data Mining*, pp. 158–174 (2002)

[Chi 05] Chi, Y., Muntz, R. R., Nijssen, S., and Kok, J. N.: Frequent Subtree Mining-An Overview, *Fundamenta Informaticae*, Vol. 66, No. 1, pp. 161–198 (2005)

[Erickson 83] Erickson, B. and Sellers, P.: Recognition of patterns in genetic sequences, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison. Addison Wesley, MA* (1983)

[Liu 03] Liu, B., Grossman, R., and Zhai, Y.: Mining data records in Web pages, in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 601–606 (2003)

[Sellers 80] Sellers, P.: The Theory and Computation of Evolutionary Distances: Pattern Recognition, *J. Algorithms*, Vol. 1, No. 4, pp. 359–373 (1980)

[Tosaka 07] Tosaka, H., Nakamura, A., and Kudo, M.: Mining Subtrees with Frequent Occurrence of Similar Subtrees, in *Proceedings of the 10th International Conference on Discovery Science*, pp. 286–290, Springer (2007)

[Zaki 05] Zaki, M.: Efficiently mining frequent trees in a forest: algorithms and applications, *Knowledge and Data Engineering, IEEE Transactions on*, Vol. 17, No. 8, pp. 1021–1035 (2005)

[Zhai 05] Zhai, Y. and Liu, B.: Web data extraction based on partial tree alignment, in *Proceedings of the 14th international conference on World Wide Web*, pp. 76–85 (2005)

[Zhang 95] Zhang, K.: Algorithms for constrained editing distance between ordered labeled trees and related problems, *Pattern Recognition*, Vol. 28, No. 3, pp. 463–474 (1995)

[Zhao 05] Zhao, H., Meng, W., Wu, Z., Raghavan, V., and Yu, C.: Fully automatic wrapper generation for search engines, in *Proceedings of the 14th international conference on World Wide Web*, pp. 66–75 (2005)